WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
l3 and l4	2

Database:

- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L6

Refine Search

Recall Text | Clear

Search History

DATE: Wednesday, December 11, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L6	l3 and l4	2	L6
L5	l3 same l4	0	L5
L4	memory near4 (sub-region or subregion or sub-section or sub-unit or sub-cell)	275	L4
L3	(clear$3 or flush$3) adj6 (memory region or memory section)	72	L3
L2	(clear$3 or flush$3) adj6 "memory region or memory section"	0	L2
L1	(clear$3 or flush$3) adj6 "memory region"	25	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 2 of 2 returned.

□ 1. Document ID: US 6219725 B1

L6: Entry 1 of 2 File: USPT Apr 17, 2001

DOCUMENT-IDENTIFIER: US 6219725 B1
TITLE: Method and apparatus for performing direct memory access transfers involving non-sequentially-addressable memory locations

Abstract Text (1):
A method and apparatus for transferring data in a computer system between a first memory region and second memory region in a single Direct Memory Access (DMA) operation. The first memory region, the second memory region, or both the first and second memory regions can include sub-regions of sequentially-addressable memory locations that are separated, within their respective regions, by a stride. The method and apparatus are particularly well adapted for use in computer graphics systems that include one or more regions of memory, such as frame buffers, that are organized in a rectangular manner as a plurality of contiguous but not sequentially-addressable memory locations within the memory of the graphics system.

Brief Summary Text (14):
It should be appreciated that although the graphics memory 250 as a whole is organized as an array of sequentially-addressable storage locations, the rectangular regions of memory 252, 254 are not; rather, each of the rectangular regions of memory 252, 254 includes sub-regions of sequentially-addressable storage locations that are separated from one another by a distance. The distance separating sub-regions of sequentially-addressable storage locations (i.e., the distance separating the first storage location in Row I from the first storage location in Row I-1) is termed the "stride" of rectangular region 252, denoted by "S", while the width of rectangular region 252 (i.e., the width of each of the sub-regions of sequentially-addressable storage locations) is denoted by "W'". This rectangular organization of memory regions 252, 254 significantly reduces the effective use of DMA controllers because conventional DMA controllers can only transfer data between source and destination locations which are both sequentially-addressable. Thus, because not all of the storage locations in rectangular regions 252, 254 are sequentially-addressable within their respective region (e.g., location N'+W'-1 and location N'+S), separate DMA operations are required for writing to or reading from each sub-region (e.g., row) of sequentially-addressable storage locations.

Brief Summary Text (18):
According to one embodiment of the present invention, a direct memory access (DMA) system for transferring data between a first memory region and a second memory region is provided. The first memory region includes a first plurality of memory locations and the second memory region includes a second plurality of memory locations, the first plurality of memory locations being separated into a plurality of sub-regions of sequentially-addressable memory locations, each of the plurality of sub-regions being separated by a stride of at least one addressable memory location. The DMA system includes a DMA controller that is configured to copy the data between each of the first plurality of memory locations and each of the second plurality of memory locations in a single DMA operation. In one aspect of this embodiment, the second memory region can be separated into a second plurality of sub-regions of sequentially-addressable sub-regions that are separated by a second stride. Moreover, in another aspect of this embodiment, the number of sub-regions in the first memory region, as well as the number of sequentially-addressable memory locations within each sub-region of the first memory region can be different than those in the second memory region.

Brief Summary Text (19):

According to another embodiment of the present invention, a method of transferring data between a first memory region and a second memory region is provided for use in a computer system. The first memory region includes a first plurality of memory locations and the second memory region includes a second plurality of memory locations, the first plurality of memory locations being separated into a plurality of sub-regions of sequentially-addressable memory locations, each of the plurality of sub-regions being separated by a stride of at least one addressable memory location. The method includes a step of copying the data between each of the first plurality of memory locations and each of the second plurality of memory locations in a single DMA operation. In one aspect of this embodiment, the step of copying the data includes steps of copying data from a first sub-regions of the plurality of sub-regions to the second memory region, advancing to a next sub-region of the plurality of sub-regions based upon the stride, and copying data from the next sub-regions to the second memory region, all in the single DMA operation.

Brief Summary Text (20):
According to a further embodiment of the present invention, method of transferring data between a first memory region and a second memory region is provided for a computer system. The first memory region includes a first plurality of memory locations and the second memory region includes a second plurality of memory locations, the first plurality of memory locations being separated into a plurality of sub-regions of sequentially-addressable memory locations, each of the plurality of sub-regions being separated by a stride of at least one addressable memory location. The method includes a step of using the stride separating each of the plurality of sub-regions to copy the data between the first memory regions and the second memory region in a direct memory access (DMA) operation. In one aspect of this embodiment, the stride can be used to copy the data between each of the first plurality of memory locations and each of the second plurality of memory locations in a single DMA operation.

Detailed Description Text (3):
Advantageously, either the source memory region, the target memory region, or both can include memory locations that are not sequentially-addressable within their respective region(s). This permits DMA operations involving non sequentially-addressable source and/or target memory regions to be performed significantly faster than using conventional DMA controllers. In particular, because embodiments of the present invention take account of the distance (i.e., the stride) that may separate sub-regions of sequentially-addressable memory locations within source and target regions of memory, DMA transfers between the source and target regions of memory can be performed in a single DMA operation.

Detailed Description Text (15):
It should be appreciated that a conventional DMA controller is necessarily limited by the smallest sequentially-addressable sub-region in either source or target regions of memory, whereas the present invention is not. That is, where both source and target regions of memory include sub-regions of memory that are not sequentially-addressable between sub-regions, any conventional DMA transfer between source and target memory regions is limited to the shortest of the source or target sub-region of sequentially-addressable memory locations. For example, where each row of the source memory region includes fifteen sequentially-addressable memory locations and each row of the target memory region includes five sequentially-addressable memory locations, any conventional DMA transfer between source and target memory regions is limited to a quantity of five memory locations (i.e., the shorter of the source and target memory sub-regions of sequentially-addressable memory locations).

Detailed Description Text (19):
Once again, although graphics memory 450, as a whole, is organized as an array of sequentially-addressable storage locations, each rectangular region 452, 454 of memory 450 includes sub-regions of sequentially-addressable storage locations (i.e., rows of memory) that are not sequentially-addressable between the sub-regions themselves (i.e., between the first storage location in row I and the last storage location in row I-1). That is, the rows of sequentially-addressable storage locations are separated from one another by a some number of memory locations (i.e., "stride"). The stride separating the first storage location in Row I of region 452 from the first storage location in Row I-1 of region 452 is again denoted by "S", and the width of rectangular region 252 is again denoted by "W'".

Detailed Description Text (33):
At step 725, the DMA controller determines whether the target memory region is in host memory or graphics memory. This can be determined from the control bit (i.e., bit 2) in

the Low Order Source address register 530 for example. In such an embodiment, when the control bit is set (e.g., one), the source memory region resides in graphics memory, and thus, the target memory region resides in host memory, and when the control bit is cleared (e.g., zero) the source memory region resides in host memory, and thus, the target memory region resides in graphics memory. When the target memory region resides in graphics memory (i.e., a transfer of data from host memory to graphics memory), steps 730-750 are performed, and when the target memory region resides in host memory (i.e., a transfer of data from graphics memory to host memory), steps 755-775 are performed. As the operation of the DMA controller is similar for both directions of data transfer, only steps 730-750 are described in detail herein.

CLAIMS:

1. A computer graphics system comprising:

a host computer including a host processor and a host memory accessible by said host processor, said host memory having a source subregion of contiguous, non-sequentially addressable source memory locations;

a graphics system responsive to said host processor for rendering images on a display device, including a graphics memory having a target subregion of contiguous, non-sequentially-addressable target memory locations equal in number to said source memory locations; and

a direct memory access (DMA) controller configured to transfer, in a single DMA operation, data stored in said memory locations of said source subregion to said memory locations of said target subregion.

11. A computer graphics system comprising:

a host processor;

a host memory accessible by said host processor, said host memory having a source subregion of contiguous source memory locations, said source subregion having a height, a width and a stride;

a graphics memory having a target subregion of contiguous target memory locations equal in number to said source memory locations, said target subregion having a height, a width and a stride,

wherein at least one of said source stride and said target stride is greater than said source width and said target width, respectively, resulting in said contiguous memory locations which are non-sequentially-addressable, and wherein one or more of said source height, source width and source stride are not equal to said target height, target width and target stride, respectively; and

a direct memory access (DMA) control system including,

a first set of one or more registers in which values representing said source height, source width, source stride and a starting address of said source subregion are stored, and

a second set of one or more registers in which values representing said target height, target width, target stride and a starting address of said target subregion are stored;

a DMA controller configured to utilize said values stored in said registers to transfer, in a single DMA operation, data stored in said memory locations comprising said source subregion to said memory locations comprising said target subregion.

19. A method for transferring data comprising the steps of:

(1) providing a host computer including a host processor and a host memory accessible by said host processor, said host memory having a source subregion of contiguous, non-sequentially-addressable source memory locations;

(2) providing a graphics system responsive to said host processor for rendering images on a display device, including a graphics memory having a target subregion of contiguous, non-sequentially addressable target memory locations equal in number to

said source memory locations;

(3) transferring, in a single DMA operation, data stored in said memory locations comprising said source subregion to said memory locations comprising said target subregion.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

□ 2. Document ID: US 6209061 B1

L6: Entry 2 of 2 File: USPT Mar 27, 2001

DOCUMENT-IDENTIFIER: US 6209061 B1
TITLE: Integrated hierarchical memory overlay having invariant address space span that inactivates a same address space span in main memory

Detailed Description Text (9):
If, by contrast, the stack regions are managed by a cache memory, the system is burdened by the requirements to maintain main memory 16 in synchronism with changed cache line entries. Accordingly, the invention allocates at least certain portions of each stack memory region (e.g., subregions 46, 48 and 50) to overlay memory 23 (see FIG. 1). By inserting the respective base pointers of the subregions (and the associated length values which define, respectively, the subregion extents) into overlay memory controller 24, addresses issued by CPU 12 onto bus 14 which fall within a subregion, are immediately recognized by overlay memory controller 24.

Detailed Description Text (10):
Thus, when CPU 12, during the execution of program 26, issues either a read or write instruction to an address value that is within the range of addresses contained in a stack subregion in overlay memory 23, overlay memory controller 24 responds to that read/write request and causes an inhibition of transmission of the read/write request to both cache controller 32 and main memory bus 20. Thus, neither wait states nor resynchronization actions are required.

Detailed Description Text (12):
It is also to be understood that further portions of stack memory regions 40, 42, and 44 can be managed by cache memory 30. However, it is clear that the most highly used regions of the stack memory region should be assigned to overlay memory 23 so as to provide the most efficient system operation. Since overlay memory 23 handles data from the most used regions of the stack, cache memory 30 is able to operate in a more effective manner since it deals with data resident in lesser used areas of the stack and main memory.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

Generate Collection | Print

Terms	Documents
l3 and l4	2

Display Format: KWIC | Change Format

Previous Page Next Page